Choose Outsiders First: a mean 2-approximation random algorithm for covering problems. *

Etienne Birmelé
Laboratoire Statistique et Génome, CNRS, Université d'Evry, France
Laboratoire de Biométrie et Biologie Evolutive, CNRS, Université Lyon 1, France
INRIA Rhône-Alpes, Montbonnot Saint-Martin, France
etienne.birmele@genopole.cnrs.fr

August 30, 2012

Abstract

A high number of discrete optimization problems, including Vertex Cover, Set Cover or Feedback Vertex Set, can be unified into the class of covering problems. Several of them were shown to be inapproximable by deterministic algorithms. This article proposes a new random approach, called *Choose Outsiders First*, which consists in selecting randomly elements which are excluded from the cover. We show that this approach leads to random outputs which mean size is at most twice the optimal solution.

In his landmark paper in complexity theory [11], R. Karp provides a list of 21 NP-complete problems from which most of the NP-completness results are deduced. Among them are the extensively studied Vertex Cover, Set Cover, Feedback Vertex (or Arc) Set or Hitting Set problems, which belong to the class of covering problems. Covering problems ask how large a certain combinatorial structure has to be to cover another one, and have a wide range of applications in all areas involving combinatorial optimization problems, including VLSI systems [10], routing [6] or scheduling [7]. In the last decades, they also became central in computational biology [12] as parsimony is often considered as the choice criteria between the different evolutional scenarios explaining the observations [9].

Most of the covering problems are NP-complete, so that they need to be solved by using heuristics. The proposed algorithms can mainly be classified into two families. The firt one consists in the primal-dual approaches which are based on the formulation of covering problems as integer linear programming problems [13]. The second type of approximation algorithms are based on local ratio techniques which consist in solving a problem locally and extending the

*This work was supported by the ERC Advanced Grant SISYPHE.

solution [2, 4]. A common measure of the quality of those heuristics is their approximation factor. The litterature about approximation results for covering problems is huge, and an overview can be found in [1]. The main covering problems listed above were shown to be APX-hard. The Set Cover is even not approximable better then within a logarithmic factor, whereas the constant approximability of Hitting Set and Directed Feedback Vertex (or Arc) Set problems are still open questions. The best known solutions for Vertex Cover and Undirected Feedback Vertex Set have an approximation ratio of 2.

One way to reach better approximation results is the use of random algorithms and the study of the mean approximation ratio of the outputs. A random local ratio approach proposed in [3] yields for instance a mean approximation of 2 for the Vertex Cover problem and of the maximum size of the sets for the Set Cover and Hitting Set problems.

In this paper, we propose a new random algorithm for covering problems. Its main difference with already studied heuristics is that the aim is not to select good candidates for the cover but to exclude randomly elements from the cover. This corresponds to assign a random order to the elements and to consider them in increasing order. An element is then added to the cover if and only if has to be added in order not to miss a structure which has to be covered. This idea was introduced in the case of the unweighted Vertex Cover in [8] and was proved to yield a mean 2-approximation for this particular covering problem [5]. We show that this approach, that we call *Choose Outsiders First*, is in fact much more general in the sense that it can be applied and yields a mean approximation ratio of 2 for any covering problem. This is to our knowledge the first approximation result for which the ratio is independent from the input for problems like the Set Cover or the Directed Feedback Vertex Set.

1 The algorithm

Following Bar-Yehuda's [3] formalism, an unweighted covering problem is a pair $(U, f : 2^U \to \{0,1\}, \omega : U \to \mathbb{R}^+)$ where U is a finite set, f is monotone, i.e., $A \subseteq B \Rightarrow f(A) \leq f(B)$, and $f(U) = 1$. For a set $C \subset U$, $\omega(C) = \sum_{x \in C} \omega(x)$ is called the weight of C. A set $C \subseteq U$ is a *cover* if $f(C) = 1$. The problem is then to find a cover of minimum weight, that is a set $C^* \subset U$ such that

$$\omega(C^*) = \min(\omega(C) : C \subset U \text{ and } f(C) = 1)$$

To do so, we consider the algorithm *Choose Outsiders First* which relies on the idea that if the optimal cover is small, a randomly chosen vertex has a high probability not to be contained in the optimal solution. Therefore, two sets OUT and IN are considered and at each step, a vertex is randomly chosen and is put into OUT, that is considered to be not in the cover. However, from time to time, a structure which has to be covered has seen all its elements but one put into OUT. This last element has then to be put into the cover and is added in the IN set. Once all the elements of U have been classified into OUT or IN, the set IN is a cover and is output by the algorithm.

The pseudo-code of *Choose Outsiders First* is given in Algorithm 1. At each step of the algorithm, we say that a vertex is *available* if it hasn't be classified yet and denote by A the set available vertices, that is $A = U \setminus \{OUT \cup IN\}$. The pseudo-code of Algorithm 1 is written by using A, IN and OUT at each step for better readability but in practice, the algorithm can be written by updating only A and IN or OUT and IN, the union of the three sets beeing always U. Note that if the conditions of Line 2 are checked in polynomial time, which is the case if the problem is in NP, the total running time is polynomial.

The probability distribution used to choose the excluded vertex at each step is the one proportional to the weights of the available vertices. Elements of small weight are therefore excluded with lower probability and thus favored to be in the output. Note that in the case of an unweighted covering problem, the algorithms picks uniformly the excluded vertex.

Algorithm 1: Choose Outsiders First

```
1 IN = ∅, OUT = ∅, A = U ;
2 while A ≠ ∅ do
3 |   Pick randomly u ∈ A with probability ω(u)/ω(A);
4 |   OUT = OUT ∪ {u} ;
5 |   for v ∈ U \ {IN ∪ OUT} such that f(U \ {OUT ∪ {v}}) = 0 do
6 |   |   IN = IN ∪ {v}
7 |   end
8 |   A = U \ {OUT ∪ IN} ;
9 end
```

The size of the output cover is a random variable, which we call $CoverSize$. To assess the efficiency of the algorithm, we have to rely the values of $CoverSize$ to the size of an optimal solution. Let us first show that this value is equal to $\min(CoverSize)$.

Theorem 1. *Any optimal cover C^* has a non-null probability to be output by* Choose Outsiders First*. Hence, the optimal size of a cover is* $\min(CoverSize)$.

Proof. Let C^* be an optimal cover. Consider a run of the algorithm such that, if possible, the random picked vertex is always chosen in $U \setminus C^*$. Let us show by induction that at each step, $OUT \cap C^* = \emptyset$ and $IN \subseteq C^*$. Note that it is trivially true at the beginning of the algorithm.

Suppose now it is true at some point just before a random vertex is picked and suppose that no vertex in $U \setminus C^*$ is available. Then $A \subset C^*$, $IN \subset C^*$ and $OUT \cap C^* = \emptyset$, that is $U \setminus OUT = C^*$. But if there is a vertex v in A, it has not been put into IN in the previous round, which means that the condition at Line 5 was not satisfied. Hence, $U \setminus \{OUT \cup \{v\}\} = C^* \setminus \{v\}$ is a cover, which contradicts the minimality of C^*. Consequently, a vertex of $U \setminus C^*$ has to be available and it is such a vertex which is chosen. Thus the two desired set relations are still valid after Line 4.

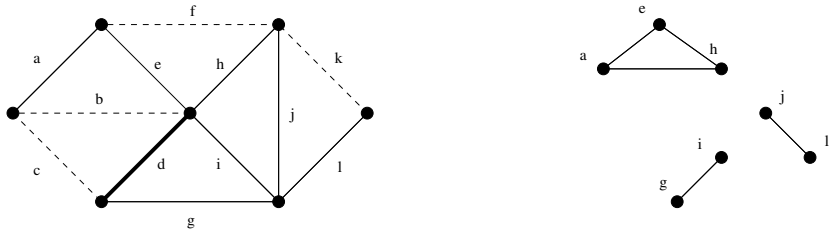

Figure 1: Consider the Feedback Arc Set problem on the graph G on the left, that is finding a set of arcs of minimal weight hitting all the cycles of G. Assume that b is picked first, followed by g, c (d is then added to IN) and h. At this point, we have $OUT = \{b, c, g, h\}$ and $IN = \{d\}$. The right part of the figure shows the resulting incompatibility graph, where two edges of G are linked if adding them both to OUT creates a cycle containing no edge in IN.

Suppose now that they are valid after Line 4 and let v an element which is added to IN at Line 6. Then v satisfied the condition on Line 5, which means that $U \setminus \{OUT \cup \{v\}\}$ is not a cover. But if $v \notin C^*$, $C^* \subseteq U \setminus \{OUT \cup \{v\}\}$, which would be a contradiction with the monotonicity of the covering property. Thus, only vertices of C^* are added to IN, so that the set relations remain true after Line 7.

□

2 Analysis of the mean approximation ratio

The key structure for the analysis of the *RANDOM COVER* algorithm is a graph encoding the fact that the choice of a vertex to put into OUT may force some others to go into IN: consider two sets OUT and IN generated by the algorithm as they are on the beginning of a run of the loop at Line 5 . We define the *incompatibility graph* $G_{OUT,IN}$ as follows:

- $V(G_{OUT,IN}) = A$
- (u, v) is an edge of $G_{OUT,IN}$ if $f(U \setminus \{OUT \cup \{u, v\}\}) = 0$.

$G_{OUT,IN}$ represents the set of elements of U which still have to be classified and two of them are linked by an edge if they are incompatible, that is both of them cannot be added simultaneously to OUT as putting all other elements in IN would not lead to a solution of the covering. Note that the incompatibility graph changes when the sets OUT and IN are updated. Moreover, if u denotes the vertex put into OUT at Line 4, the set of vertices put into IN at Line 6 is exactly the neighborhood $N(u)$ of u in $G_{OUT,IN}$.

An example of incompatibility graph is shown in Figure 1 in the context of a Feedback Arc Set problem.

Consider again any pair (OUT, IN) of sets generated by the algorithm. Let $X_{OUT,IN}$ be the random variable counting the weight of the elements of U which will be added in the future to IN. The weight of the elements already in IN is not counted here. In particular, $X_{OUT,IN} = 0$ if $OUT \cup IN = U$ and $CoverSize = X_{\emptyset,\emptyset}$.

Proposition 1 can easily be adapted to show that the minimum weight of the vertices to add to IN in order to obtain a cover containing all the vertices of IN and none of OUT is $\min(X_{OUT,IN})$.

Lemma 2. *Let $G_{OUT,IN}$ be an incompatibility graph and S the vertices corresponding to a minimum solution, that is such that $IN \cup S$ is a cover and $\omega(S) = min(X_{OUT,IN})$. For any vertex u of G, denote by $N^S(u)$ the set of its neighbors in S. Then:*

1. *the set H of the vertices which are not in S is an independent set.*
2. *for every vertex u,*

$$\min(X_{OUT\cup\{u\},IN\cup\{N(u)\}}) \leq \min(X_{OUT,IN}) - \sum_{v\in N^S(u)} \omega(v)$$

Proof.

1. Suppose that an edge links to vertices u and v of H. It means that $U \setminus \{OUT \cup \{u, v\}\}$ is not a cover, which, together with the fact that $IN \cup S$ is a cover, contradicts the monotonicity of the covering property.
2. When u is added to OUT, the whole neighborhood of u is added to IN. In particular, all the vertices of $S \cap N(u)$ are added to IN. Hence, starting from $OUT' = OUT \cup \{v\}$ and $IN' = IN \cup N(u)$, it is possible to complete IN' into a cover by adding the vertices of $S \setminus N(u)$. The optimal solution is therefore of weight at most $\omega(S) - \sum_{v\in N^S(u)} \omega(v)$.

□

Theorem 3. *For all pair of sets OUT and IN that may be generated by the algorithm,*

$$\mathbb{E}(X_{OUT,IN}) \leq 2\min(X_{OUT,IN})$$

In particular, applying it for $OUT = \emptyset$ and $IN = \emptyset$ yields

$$\mathbb{E}(CoverSize) \leq 2\min(CoverSize)$$

Proof. The proof is done by induction on $|A|$.

If $|A| = 0$, $X_{OUT,IN}$ is constant and equal to 0 so that the theorem trivially holds.

Let's consider a pair (OUT, IN) generated by the algorithm and suppose that the theorem holds for every pair (OUT', IN') with $OUT \subset OUT'$ and $IN \subset IN'$.

To improve readability, the indices OUT and IN are omitted in the rest of this proof: X (resp. G) stands for $X_{OUT,IN}$ (resp. $G_{OUT,IN}$) and $X_{+u,+N(u)}$ for $X_{OUT\cup\{u\},IN\cup N(u)}$.

As in Lemma 2, S is a optimal size solution given OUT and IN and N, N^S and N^H stand for the different neighborhoods in the incompatibility graph $G_{OUT,IN}$.

$$\begin{aligned}
\mathbb{E}(X) &= \sum_{uinA} \mathbb{E}(X|u \text{ is chosen })\mathbb{P}(u \text{ is chosen }) \\
&= \frac{1}{\omega(A)} \sum_{u\in A} \omega(u)\mathbb{E}(X|u \text{ is chosen }) \\
&= \frac{1}{\omega(A)} \sum_{u\in A} \omega(u)\mathbb{E}(X_{+u,+N(u)}) + \sum_{v\in N(u)} \omega(v) \\
&\leq \frac{1}{\omega(A)} \sum_{u\in A} \omega(u)\big(2\min(X_{+u,+N(u)}) + \sum_{v\in N(u)} \omega(v)\big) \qquad \text{by induction} \\
&\leq \frac{1}{\omega(A)} \sum_{u\in A} \omega(u)\big(2(\min(X) - \sum_{v\in N^S(u)} \omega(v)) + \sum_{v\in N(u)} \omega(v)\big) \qquad \text{by Lemma 2} \\
&\leq 2\min(X) + \frac{1}{\omega(A)} \sum_{u\in A} \omega(u)\big(-2\sum_{v\in N^S(u)} \omega(v) + \sum_{v\in N(u)} \omega(v)\big) \\
&\leq 2\min(X) + \frac{1}{\omega(A)} \sum_{u\in A} \omega(u)\big(\sum_{v\in N^H(u)} \omega(v) - \sum_{v\in N^S(u)} \omega(v)\big) \qquad (1)
\end{aligned}$$

For any edge $e = (u, v)$ of the incompatibility graph, we define its weight as the product of the weight of its endvertices, that is $\omega(e) = \omega(u)\omega(v)$. Let $e(H, S)$ denote the total weight of the edges linking S to H, that is $e(H, S) = \sum_{e=(u,v),u\in H,v\in S} \omega(e)$.

Then, as H is an independent set,

$$\begin{aligned}
\sum_{u\in H} \omega(u)\big(\sum_{v\in N^H(u)} \omega(v) - \sum_{v\in N^S(u)} \omega(v)\big) &= -\sum_{u\in H} \omega(u)\big(\sum_{v\in N^S(u)} \omega(v)\big) \\
&= -e(H, S)
\end{aligned}$$

and

$$\begin{aligned}
\sum_{u\in S} \omega(u)\big(\sum_{v\in N^H(u)} \omega(v) - \sum_{v\in N^S(u)} \omega(v)\big) &\leq \sum_{u\in S}\sum_{u\in S} \omega(u)\big(\sum_{v\in N^H(u)} \omega(v)\big) \\
&\leq e(H, S)
\end{aligned}$$

Thus, Equation 1 yields

$$\mathbb{E}(X) \leq 2\min(X),$$

which proves the theorem.

□

Using the standard Markov Inequality, this theorem allows to obtain almost surely a $2+\alpha$ approximation for every positive α as stated in the following corollary.

Corollary 4. *Consider any covering problem in NP. For every $\alpha > 0$ and $\epsilon > 0$, there exist a polynomial time random algorithm which output is a $2+\alpha$ approximation with probability at least $1-\epsilon$.*

Proof. Consider one run of the *Choose Ousiders First* algorithm. Let X be the weight of the output and Opt be the weight of an optimal solution. Then

$$\begin{aligned}
\mathbb{P}(X > (2+\alpha)Opt) &\leq \frac{\mathbb{E}(X)}{(2+\alpha)Opt} && \text{by Markov's inequality} \\
&\leq \frac{2Opt}{(2+\alpha)Opt} && \text{by Theorem 3} \\
&\leq \frac{1}{1+\alpha/2}
\end{aligned}$$

Thus, running the algorithm p times with $p \geq \frac{-\ln \epsilon}{\ln(1+\alpha/2)}$ and taking the minimum X^* among all the outputs yields

$$\begin{aligned}
\mathbb{P}(X^* > (2+\alpha)Opt) &\leq \big(\frac{1}{1+\alpha/2}\big)^p \\
&\leq \epsilon
\end{aligned}$$

□

Acknowledgements I want to thank Marie-France Sagot, Pierluigi Crescenzi, Alberto Marchetti-Spaccamela and Leen Stougie for useful comments.

References

[1] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and Approximation: Combinatorial optimization problems and their approximability properties.* Computer Science Series. Springer Verlag, 1999.

[2] V. Bafna, P. Berman, and T. Fujito. Constant ratio approximation of the weighted feedback vertex set problem for undirected graphs. In *ISAAC 95 Algorithms and Computation*, volume 2, pages 142–151, 1995.

[3] R. Bar-Yehuda. One for the price of two: a unified approach for approximating covering problems. *Algorithmica*, 27:131–144, 2000.

[4] R. Bar-Yehuda and S. Even. A local-ratio theorem for approximating the weighted vertex cover problem. *Annals of Discrete Mathematics*, 27:27–46, 1985.

[5] E. Birmelé, F. Delbot, and C. Laforest. Mean analysis of an online algorithm for the vertex cover problem. *Information Processing Letters*, 109(9):436–439, 2009.

[6] J. Bramel and D. Simchi-Levi. The vehicle routing problem. chapter Set-covering-based algorithms for the capacitated VRP, pages 85–108. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2001.

[7] A. Caprara, M. Fischetti, and P. Toth. A heuristic method for the set covering problem. *Operation Research*, 47:730–743, 1999.

[8] F. Delbot and C. Laforest. A better list heuristic for vertex cover. *Information Processing Letters*, 107:125–127, 2008.

[9] J. Hein. Reconstructing evolution of sequences subject to recombination using parsimony. *Mathematical Biosciences*, 98(2):185–200, 1990.

[10] D. Hochbaum and W. Maass. Approximation algorithms for covering and packing problems in image processing and vlsi. *J. Assoc. Comp. Mach.*, 32:130–136, 1985.

[11] R. Karp. Reducibility among combinatorial problems. In R. Miller and J. Thatche, editors, *Complexity of Computer Computations*, pages 85 – 103. Plenume Press, 1972.

[12] J.C. Setubal and J. Meidanis. *Introduction to Computational Molecular Biology*. Computer Science Series. PWS Pub., 1997.

[13] V. Vazirani. *Approximation algorithms*. Springer, 2001.